**Report Buildroot:**

1. **Buildroot:**

**1.1 - Introduction:**

- Buildroot is a toolkit for building highly customized embedded Linux images.

- Buildroot is a tool that simplifies and automates the process of building a complete Linux system for an embedded system, using cross-compilation.

- Buildroot is able to generate a cross-compilation toolchain, a root filesystem, a Linux kernel image and a bootloader for your target.

Why choose buildroot?

- The main advantage of Buildroot is that it is easy to learn, especially as it uses normal Makefiles and the Kconfig language.

- Buildroot easy to use and automates.

- Easily select packages to install.

- Buildroot is useful mainly for people working with embedded systems. Embedded systems often use processors that are not the regular x86 processors everyone is used to having in his PC. They can be PowerPC processors, MIPS processors, ARM processors, etc.

**1.2 - Prerequisites:**

- Host workstation run Linux ( Ubuntu or VM as the host).

- 15GiB free disk space

- 2-4 hours free time

- Working knowledge of Linux

* 1. **– Process:**

**There’s four main steps:**

• Install host tools:

+ Git: để lấy tài liệu từ github

+ A compiler:  an initial compiler to build its own compilers

( make binutils build-esential gcc g++) và các công cụ dùng cho buildroot:

sudo apt install -y git build-essential wget cpio unzip rsync bc libncurses5-dev screen

• Dowload Buildroot:

git clone git://git.buildroot.net/buildroot

or git clone <https://git.buildroot.net/buildroot>

Lưu ý: chọn phiên bản phù hợp để không bị “Legacy Configurations”

• Configure Buildroot

+ Configure Buildroot for your target, for each board have each config

+ The exact command to do this depends on which board you’re building

• Build the target image

1. **How Buildroot Work :**

**2.1 Để hiểu được quá trình hoạt động của buildroot cần phải nắm rõ 2 thứ:**

* **Buildroot packages**

+ Most things in Buildroot are packages. You can see these packages under the package directory in the Buildroot tree.

+ Each package defines its own configuration options, build steps, and dependencies. The dependencies tell Buildroot in what order to build packages, the build steps tell exactly what commands to run to build the package (including how to download its source code), and the configuration options control anything about the package.

+ The package’s list of configuration options is in a file called Config.in using a language called [Kconfig](https://www.kernel.org/doc/Documentation/kbuild/kconfig-language.txt)

* **Build configurations**

+ Taken together, all the configuration options for all a system’s packages form a build configuration. Applying one of these default build configurations—a “defconfig”—allows you to quickly select all the relevant options.

+ And so this is how make knows how to build everything: all the configuration options, for all the packages, were supplied in the raspberrypi0w\_defconfig file that we specified at the beginning of the build.6 (GNU Make calculates the dependency tree and starts building the packages in the correct order.)

+ Selecting the defconfig copies all the config options into the working configuration kept in a file named .config. You can change the working configuration using the menuconfig tool. This provides a graphical interface that lets you find and change various menu options.

+ buildroot check và sẽ dựa vào các packages mà bạn chọn lựa trong option configuration của target mà bạn lưu lại ( .config) để tải về từ internet và compile lại theo thứ tự.

**2.2 When you understand buildroot packages and build configurations. Now there’s how buildroot work:**

Diagram

Description automatically generated

1. Buildroot builds the toolchain, which is the cross-compilers and other tools that it needs to compile the target system.
2. The source code for each piece of software is downloaded from the Internet.
3. Check each option in Config.in to compile and dowload from Internet
4. Using Buildroot scripts, source is unpacked, patched, configured, compiled, and installed into the target output directory that forms the root filesystem (“rootfs”) for the target.
5. Extra files, such as on-device configuration files, also get copied into the target output directory.
6. Finally, scripts assemble the final firmware image from this rootfs.

There are some exceptions; sometimes the cross-compiler is downloaded without needing to be compiled. Sometimes the manufacturer ships an entire “Board Support Package” (BSP) where all the compilation has already been done for you. But at the end of the day, those just truncate this flowchart; all the steps still have to be accomplished somehow.

Here are the most important directories in the top of the Buildroot tree:

|  |  |
| --- | --- |
| Directory | Purpose |
| board/ | Files and scripts to support each target board |
| configs/ | Build configurations |
| packpage/ | Package definitions |
| output/host/ | Build tools that run on the workstation |
| output/target/ | Target output directory where target binaries are staged |
| output/images/ | Filesystem images and the final firmware images are emitted here |

1. **Buildroot build the Licheepi\_Nano images:**

**3.1: Build the buildroot config**

First, clone this repo:

git clone [git@github.com:unframework/licheepi-nano-buildroot.git](mailto:git@github.com:unframework/licheepi-nano-buildroot.git)

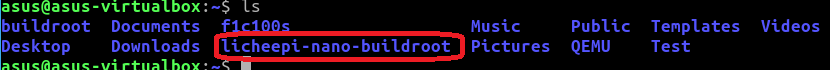
if we encounter the issue of the public key as shown in Fig, we should use

git clone https://github.com/unframework/licheepi-nano-buildroot

Ensure the specific build scripts are executed

chmod a+x licheepi-nano-buildroot/board/licheepi\_nano/\*.sh

Ensure the buildroot version used to build the .config, in this case buildroot 2020.2



**3.2: Build the buildroot**

\*\*Should choose your home folder where you don’t need to use sudo

Tải buildroot:

git clone git://git.buildroot.net/buildroot

or git clone https://git.buildroot.net/buildroot

when buildroot is clone to local, we should choose a right branch name, which is the version to build .config in step 1. In this case, we have

cd buildroot # the clone buildroot repo in local machin

git checkout 2020.02

Text

Description automatically generated

git branch -a

Git tells us we aren’t on a branch (“detached HEAD”). That’s okay for now.

Để hiểu rõ buildroot và quá trình chạy buildroot:

<https://www.thirtythreeforty.net/posts/2020/01/mastering-embedded-linux-part-3-buildroot/>

Everything should build on the buildroot source code that we have just clone. We don’t build on the buildroot having the .config file since it is just the build result and the pre-configuration .config that we have to do it alone with the command “make menuconfig” later. Thus we change directory into buildroot

cd buildroot

Cách 1: Tùy vào máy có thể được hay không

then we define that buildroot using an external buildroot source as

BR2\_EXTERNAL= /home/asus/licheepi-nano-buildroot make licheepi\_nano\_defconfig

where /home/asus/licheepi-nano-buildroot is the folder that we have just clone the pre-configuation .config. It means that we decide to use all configuration of buildroot, including Kernel, toolchain in /home/asus/licheepi-nano-buildroot. Then we should check with

make menuconfig

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

to ensure the current version of buildroot 2020.02 does make any “legacy configurations”. If everything goes well, we build all project

make

To start building:

• download source files (as required);

• configure, build and install the cross-compilation toolchain, or simply import an external toolchain;

• configure, build and install selected target packages;

• build a kernel image, if selected;

• build a bootloader image, if selected;

• create a root filesystem in selected formats

it should take more than 1 hour and a half to built and the output is the image to burn into sd card. It is located at “output/image/sdcard.img”

Cách 2: Bởi vì trong file buildroot gốc không có defconfig của licheepi\_nano

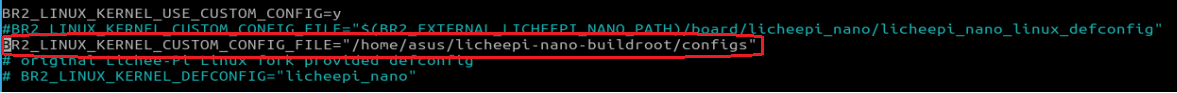


Cho nên ta copy file licheepi\_nano\_defconfig trong /home/asus/licheepi-nano-buildroot/configs/ vào /home/asus/buildroot/configs/

Text

Description automatically generated

Để kiểm tra đường dẫn đúng hay sai:



Then

make licheepi\_nano\_defconfig

Có thể chỉnh lại lựa chọn trong make menuconfig để xem lại các option cơ bản của target.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

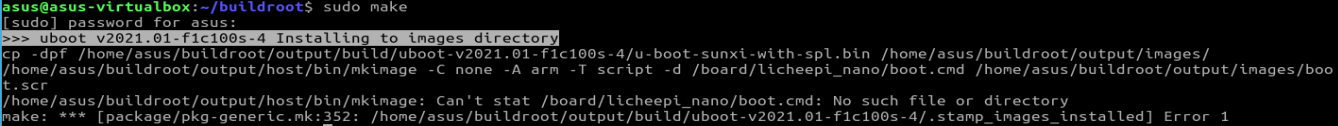
Description automatically generated

make

or sudo make

it should take more than 1 hour and a half to built and the output is the image to burn into sd card. It is located at “output/image/sdcard.img”

\*\*Khi sử dụng cách 2 nếu các gặp lỗi sau:



Cách khắc phục:

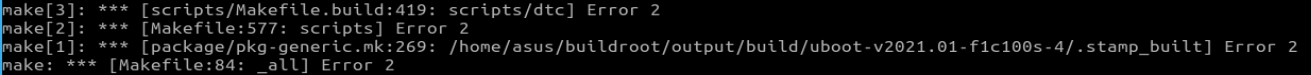
BR2\_EXTERNAL= /home/asus/licheepi-nano-buildroot make licheepi\_nano\_defconfig

where /home/asus/licheepi-nano-buildroot is the folder that we have just clone the pre-configuation .config. It means that we decide to use all configuration of buildroot, including Kernel, toolchain in /home/asus/licheepi-nano-buildroot

then

make

Hoặc lỗi như sau:



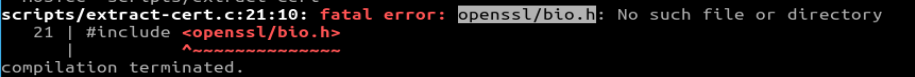
Cách khắc phục:

sudo apt-get install swig python-dev python3-dev

sudo apt-get install pkg-config

sudo apt-get install libusb-1.0-0-dev zlib1g-dev

Hoặc lỗi như sau:



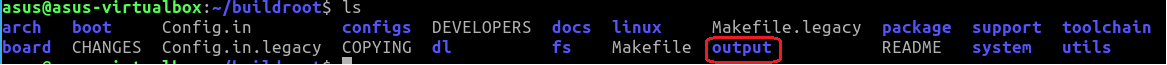
Cách khắc phục:

sudo apt-get install libssl-dev

Kết quả khi buildroot hoàn tất:

Text

Description automatically generated





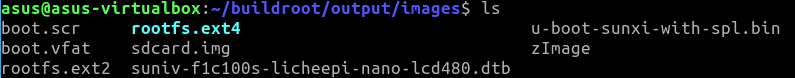
+ Host side: output/host



* Cross compiler name: “arm-buildroot-linux-gnueabi”
* Cross compiler binaries: output/host/bin

Graphical user interface

Description automatically generated



+ Root file system: rootfs.ext2 or rootfs.ext4

+ Kernel: zImage

+ Device tree: suniv-f1c100s-licheepi-nano-lcd480.dtb

Để ý trong khi buildroot sẽ có quá trình u-boot:

Text

Description automatically generated

Text

Description automatically generated

Reference:

<https://www.thirtythreeforty.net/posts/2020/01/mastering-embedded-linux-part-3-buildroot/>

<https://github.com/unframework/licheepi-nano-buildroot>